ディープラーニングを用いた AI

研究者：林

１　はじめに

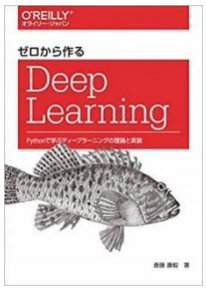
３年の課題研究で少し新しいことに挑戦してみたくて、興味があるディープラーニングに挑戦しようと思った。

２　研究の内容

プログラミング言語 Python によるディープラーニングを様々な環境で行う。

３　研究過程

４～８月　　：Python とディープラーニングの調べ学習
９～１１月　：プログラム作成
１１、１２月：学習の高速化の検討 他機材で環境構築
１月　　：比較と資料作成



↑Python 学習に用いた文献

４　使用機器

PC1：IT-PC14(電子計算機室の PC) プロセッサ　intel(R) Core(TM)2 Duo CPU E8400 @3.00GHz RAM 4.00GB OS Windows7 Pro 32 ビット
PC2：DESKTOP-DDBLGF3 プロセッサ intel(R) Core(TM) i7-6567U CPU @3.30GHz RAM 8.00GB OS Windows10 Home 64 ビット
PC3：DESKTOP-KBS7VVO プロセッサ intel(R) Core(TM) i7-6700K CPU @4.00GHz RAM 16.00GB OS Windows10 Home 64 ビット GPU NVIDIA GeForce GTX 960
タブレット：d-02K CPU Hisilicon Kirin 659 RAM 3.00GB OS Android 8.0.0

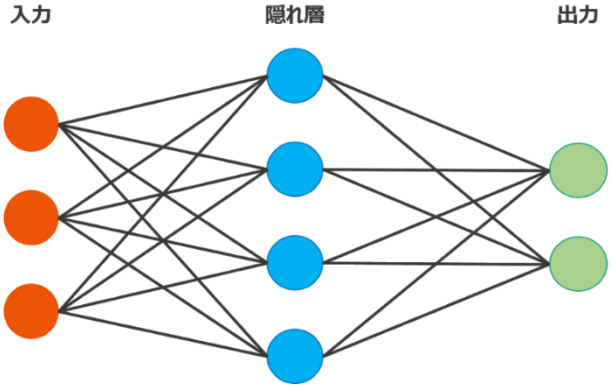
５　研究成果

（１）　MNIST による機械学習

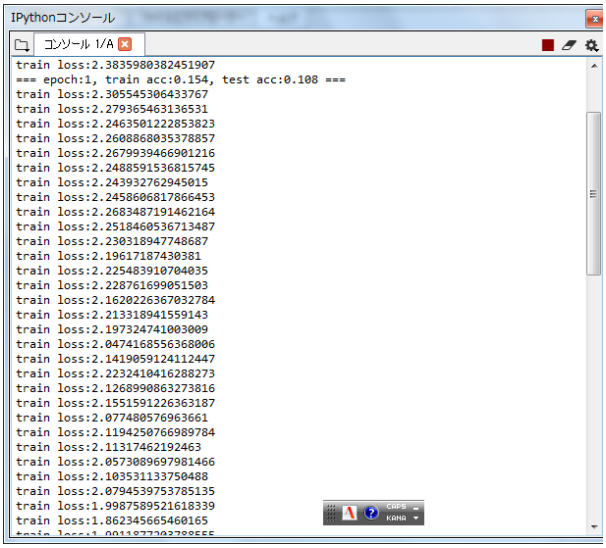
ニューラルネットワークを組み立て、手書き文字の画像(MNIST)と文字データを読み込んで機械学習をし、文字認識をする。

以下のプログラムで学習を実行する。

```
#使用するライブラリのインポート
import numpy as np
import matplotlib.pyplot as plt
from dataset.mnist import load_mnist
from deep_convnet import DeepConvNet
from common.trainer import Trainer

#MNIST のダウンロード
(x_train,t_train),(x_test,t_test)=load_mnis
t(flatten=False)

#ニューラルネットワークの接続
network = DeepConvNet()
#1 度に 100 個のデータを使って 20 回学習させる
用に設定し、実行
trainer = Trainer(network, x_train, t_train,
                  x_test,t_test,epochs=20,m
                  ini_batch_size=100,optimi
                  zer='Adam',optimizer_para
                  m={'lr':0.001},evaluate_s
                  ample_num_per_epoch=1000)
trainer.train()

# パラメータの保存
network.save_params("deep_convnet_params2.p
kl")
print("Saved Network Parameters!")
```



↑ニューラルネットワークのイメージ図



↑学習中の結果ログ
↑学習終了時「Final Test Accuracy」と表示される

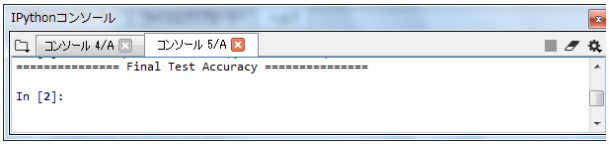
（２） 各機器による学習の機能差

学習を行う機器によってどれくらい学習の時間に差が出てくるのかを調べる。まず PC1 と PC2 を上記と同じプログラムで実行してみる。

結果

コンピュータ	学習時間
PC1	およそ 15 時間
PC2	およそ 10 時間

よって PC1 と PC2 の間には 1.5 倍ほどの時間差がある。そして、PC2 と PC3 だが、上記のプログラムだと、時間がかかりすぎてしまうため、



以下のプログラムを実行する。

```
#ライブラリのインポート
import tensorflow as tf
mnist = tf.keras.datasets.mnist

#MNIST をロード
(x_train,y_train),(x_test,y_test)=mnist.loa
d_data()
x_train,x_test=x_train/255.0,x_test / 255.0

#ニューラルネットワークを生成
model = tf.keras.models.Sequential([
        tf.keras.layers.Flatten(),
        tf.keras.layers.Dense(512,
        activation=tf.nn.relu),
        tf.keras.layers.Dropout(0.2),
        tf.keras.layers.Dense(10,
        activation=tf.nn.softmax)
        ])
#ネットワークをコンパイル
model.compile(optimizer='adam',loss='sparse
_categorical_crossentropy',metrics=['accura
cy'])

#学習回数 5 回に設定
model.fit(x_train, y_train, epochs=5)
#学習を評価
model.evaluate(x_test, y_test)
```

上記のプログラムは Python のライブラリの Tensorflow を用いた学習プログラムである。これにより、PC1 と PC2 で行ったプログラムより早く学習される。しかし、ネットワークの構成は殆ど同じである。

結果

コンピュータ	学習時間
PC2	1 分 20 秒
PC3	26 秒

こちらでは、約 3 倍の学習時間に差が出ている。よって PC1 と PC3 には、4.5 倍もの差が出ていることになる。このことより、学習時間は GPU >CPU >OS の順で、学習時間の差が出てくることが考えられる。

６　まとめ

（１）成果

今まで基本的に C 言語しか触ってこなかったものの、そこから急に Python を学ぶことにしたので、学習に時間がかかったものの,Python を学ぶことができた。

Python を使ってニューラルネットワークの仕組みやディープラーニングについて学べた。

（２）課題

当初は、パートナーの PC を使って研究を進めていく予定だったが、9 月に入ってから、パートナーが来なくなってしまい、やろうとしたことがほぼ全てできなくなってしまい、完全に他人依存だったなと反省した。

もっと作業の幅が増やせるように、予め家の PC を遠隔で操作できる方法なども早急に思いつき、調べるべきだった。

７　感想

私が興味を持っているディープラーニングや画像処理について触れられたのはよかったと思う。だが、私がやりたかったカメラによる画像認識がしっかりできなかったことが心残りである。